

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине
‘Функциональная схемотехника’

Вариант №6

Выполнил:
Студент группы Р33312
Соболев Иван
Александрович
Преподаватель:
Табунщик Сергей
Михайлович



УНИВЕРСИТЕТ ИТМО

Санкт-Петербург, 2024

Оглавление

Цели работы:.....	2
Задание:	2
Часть №1. LTSpice.....	2
Разработка вентиля.	2
Разработка БОЭ.	5
Часть № 2. Verilog.	12
Выводы по работе:	18

Цели работы:

1. Получить базовые знания о принципах построения цифровых интегральных схем с использованием технологии КМОП.
2. Познакомиться с технологией SPICE-моделирования схем на транзисторах.
3. Получить навыки описания схем базовых операционных элементов (БОЭ) комбинационного типа на вентиляльном уровне с использованием языка описания аппаратуры Verilog HDL.

Задание:

Логический базис – NAND; БОЭ – позиционный дешифратор «3 в 8».

Часть №1. LTSpice.

Разработка вентиля.

Схема разработанного вентиля:

VIN{1, 2} – входы, VOUT - выход, VDD – напряжение питания;
использовано по 2 транзистора PMOS и NMOS.

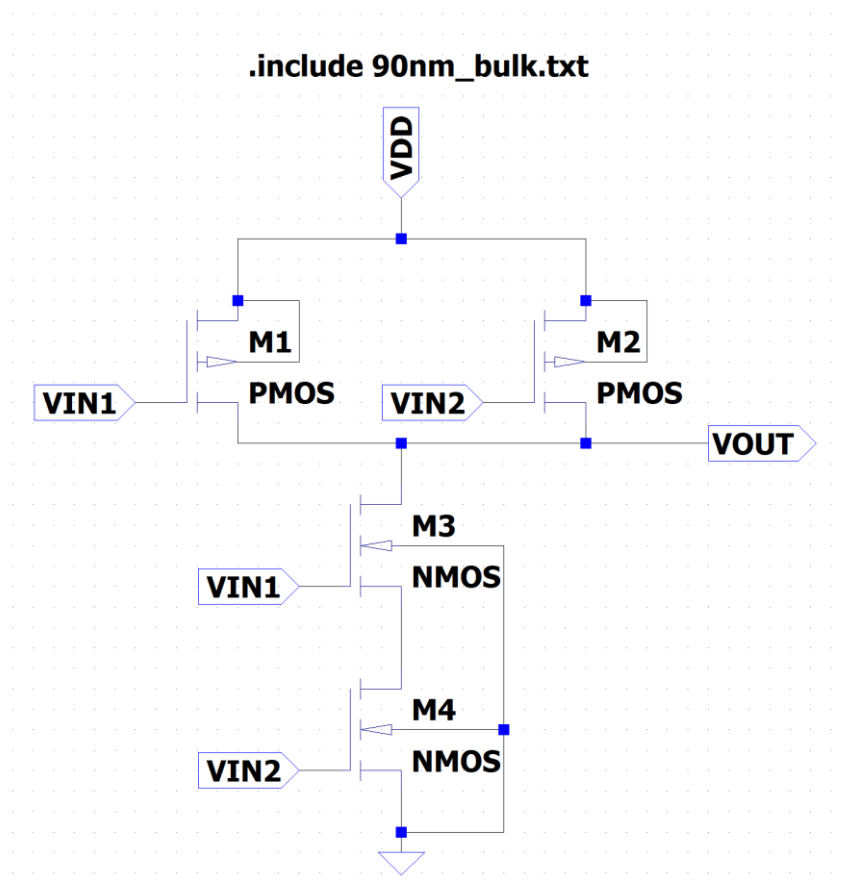


Рисунок 1 – схема вентиля

Символ вентиля:

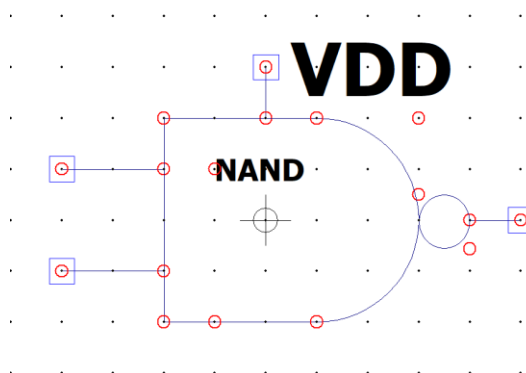


Рисунок 2 – символ вентиля

Схема тестирования:

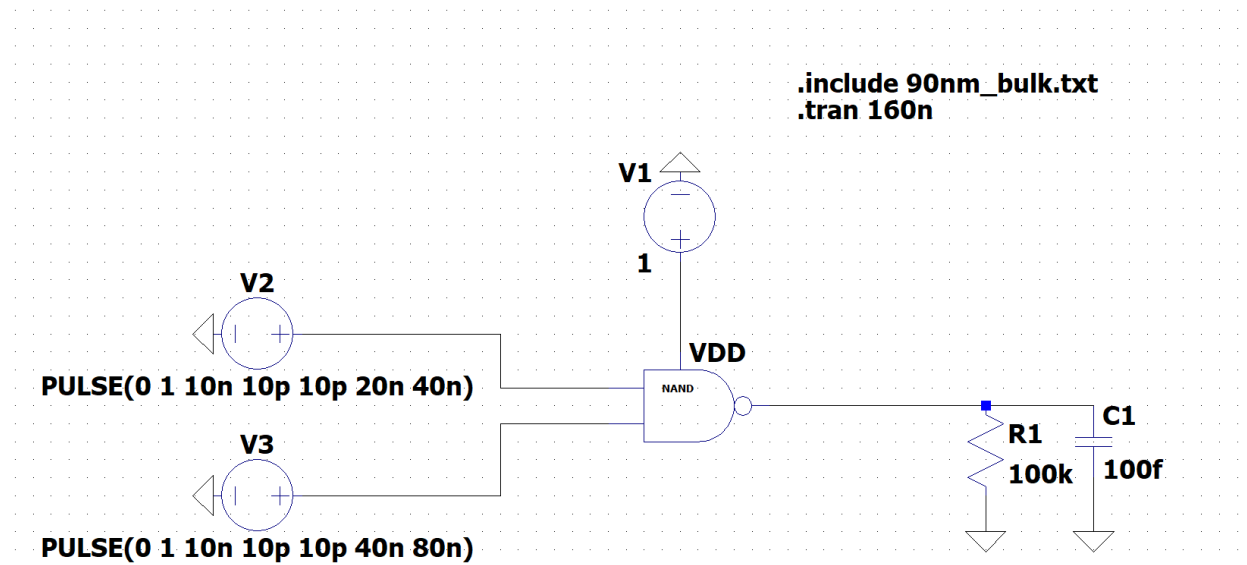


Рисунок 3 – схема тестирования

VDD = VIN = 1 В; начальное напряжение - 0 В, активное напряжение - 1 В, задержка запуска - 10 нс, время фронта и спада - 10 пс, активное время первого источника напряжения - 20 нс, период - 40 нс, для следующего последние две характеристики в два раза больше, а частоты, соответственно, меньше; резистор и конденсатор отвечают за имитацию задержки.

Временная диаграмма процесса тестирования вентиля:

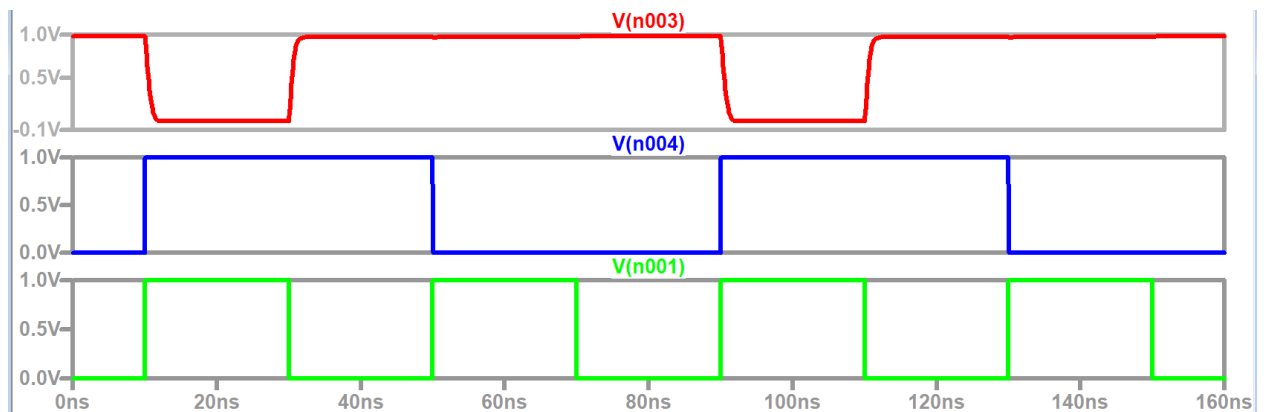


Рисунок 4 – временная диаграмма NAND

1 В – логическая единица, 0 В – логический ноль; NAND подаёт на выход единицу во всех случаях, кроме равенства единице обоих входных сигналов.

Это отображено на рисунке – на 10 и 90 нс выходной сигнал падает в ноль и остаётся таким в течение 10 нс, после чего выходит обратно в единицу.

Результат измерения задержки распространения сигнала через клапан:

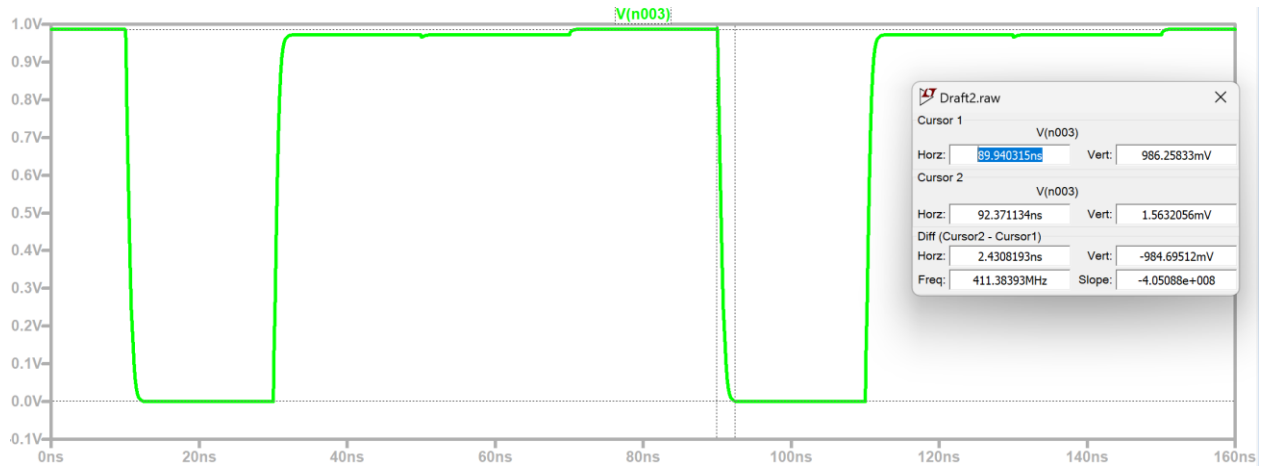


Рисунок 5 – задержка спада

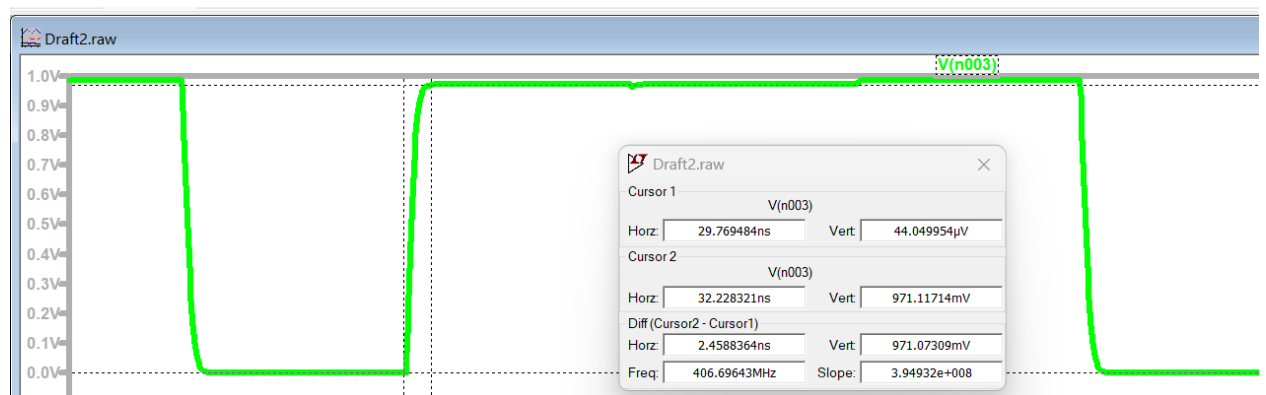


Рисунок 6 – задержка фронта

Два курсора на верхней (~ 1 В) и нижней (~ 0 В) границах заднего фронта.

Задержка фронта равна $T1 \sim 2,46$ нс

Задержка спада равна $T2 \sim 2,43$ нс

Тогда максимальная частота работы клапана равна $f = 1/(T1+T2) = 204$ МГц

Разработка БОЭ.

На базе данного клапана для удобства разработки БОЭ (дешифратора 3 в 8) я создал также инвертор и NAND с 4 входами. Инвертор необходим для конечного инвертирования. Так как обычно схема инвертора состоит из AND, а мы используем NAND – надо еще раз инвертировать.

Схема и символ инвертора.

Логическое выражение: $\underline{A} = A \& A$

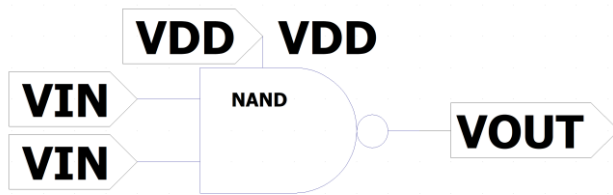


Рисунок 7 – схема инвертора

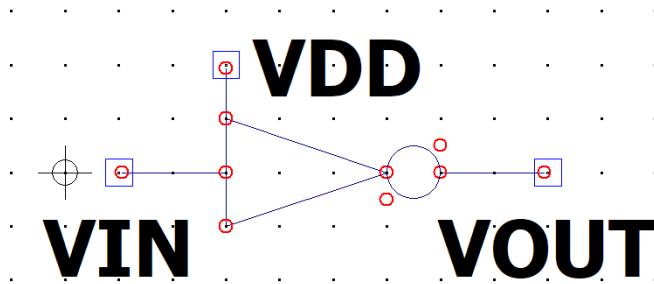


Рисунок 8 – символ инвертора

Схема и символ NAND с 4 входами.

Логическое выражение: $\underline{\underline{A \& B \& C \& D}} = \underline{\underline{A \& B \& C \& D}}$

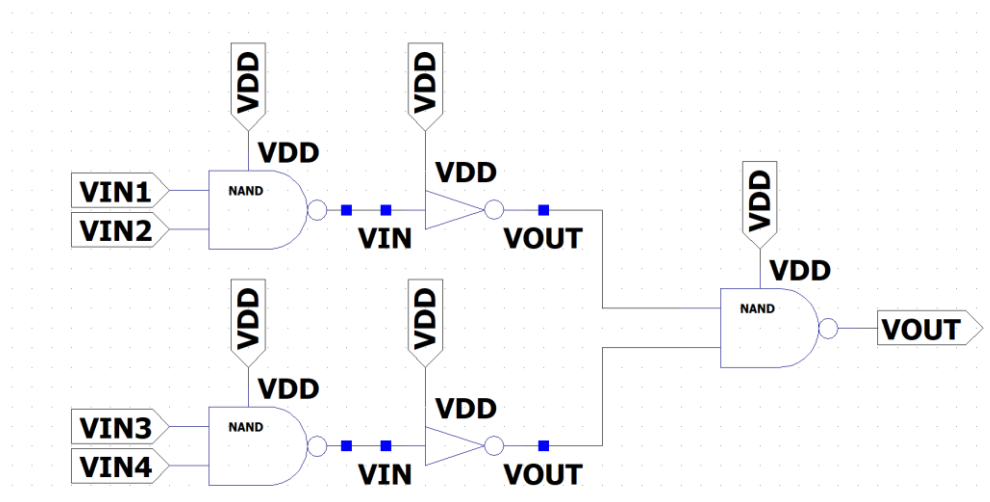


Рисунок 9 – схема NAND с 4 входами

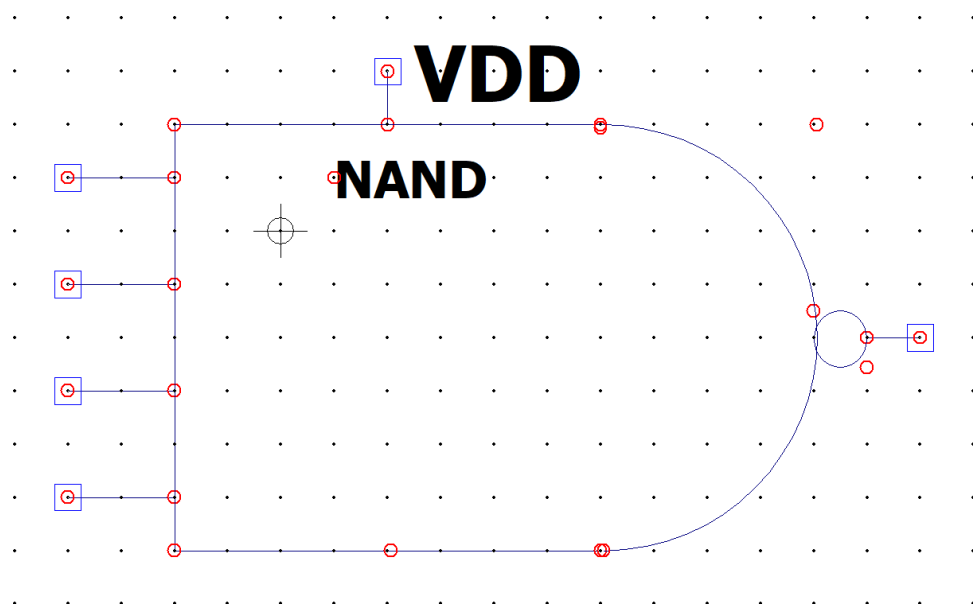


Рисунок 10 – символ NAND с 4 входами

Временная диаграмма процесса тестирования вентиля:

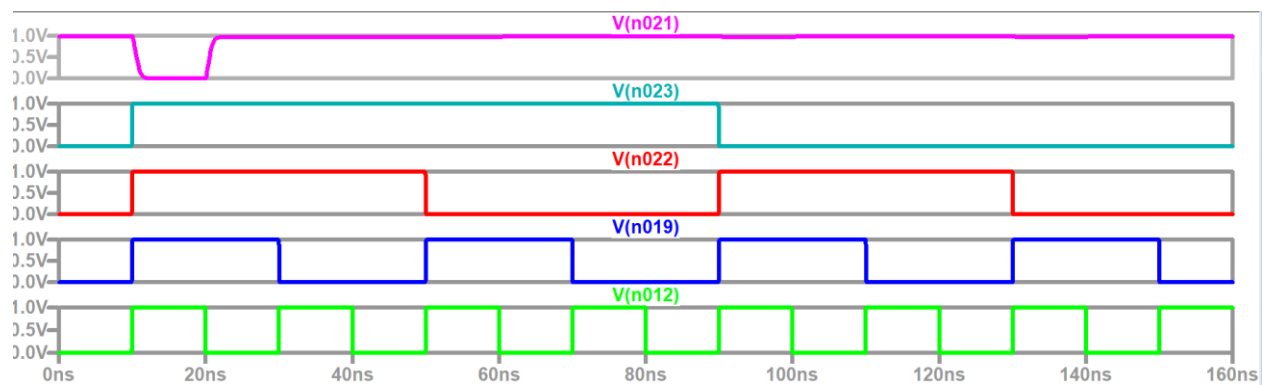


Рисунок 11 – временная диаграмма тестирования NAND с 4 входами

Схема разработанного БОЭ:

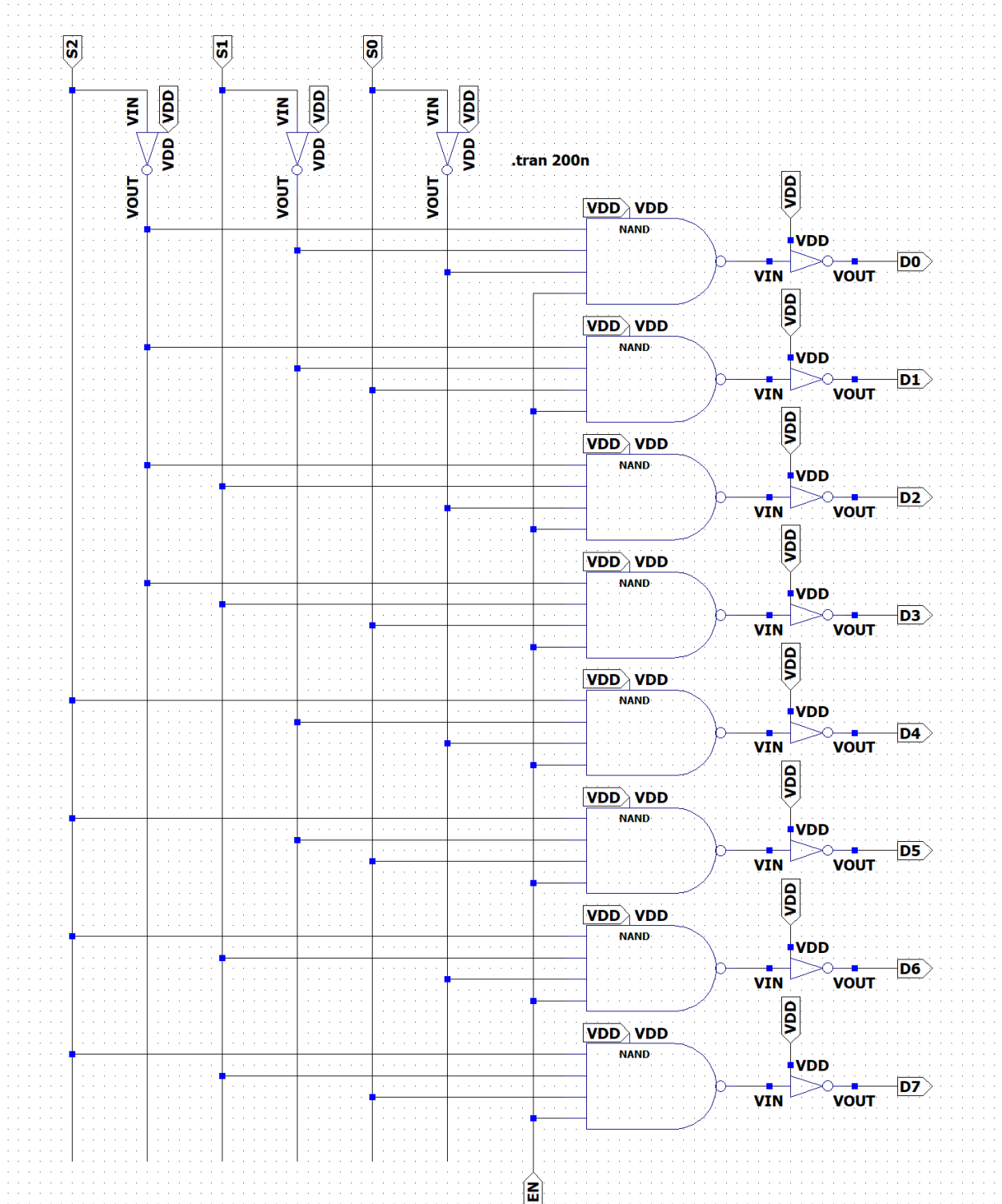


Рисунок 12 – схема БОЭ

Модуль преобразует трёхразрядное двоичное число на входе в десятичное на выходе (вернее, в его унитарный код). На вход модулю подаётся двоичное число от 0 до 7 (входы $S\{0-2\}$ соответствуют его разрядам, S_2 - старший, S_0 - младший), на выходе находятся 8 бит ($D\{0-7\}$), один из которых, порядок которого соответствует входу, становится равным 1, если сигнал разрешения EN (Enable), отвечающий за активность дешифратора, равен 1 (при равном 0

весь выход тоже будет нулевой). При изменении входа единице станет равен уже другой бит, а предыдущий обнулится.

Символ разработанного БОЭ:

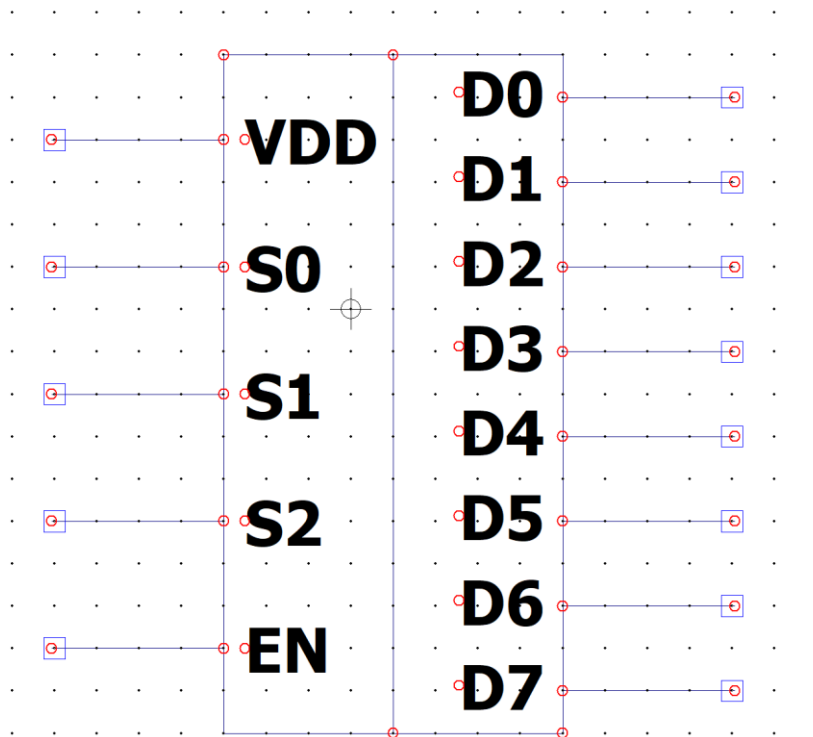


Рисунок 13 – символ БОЭ

Схема тестирования:

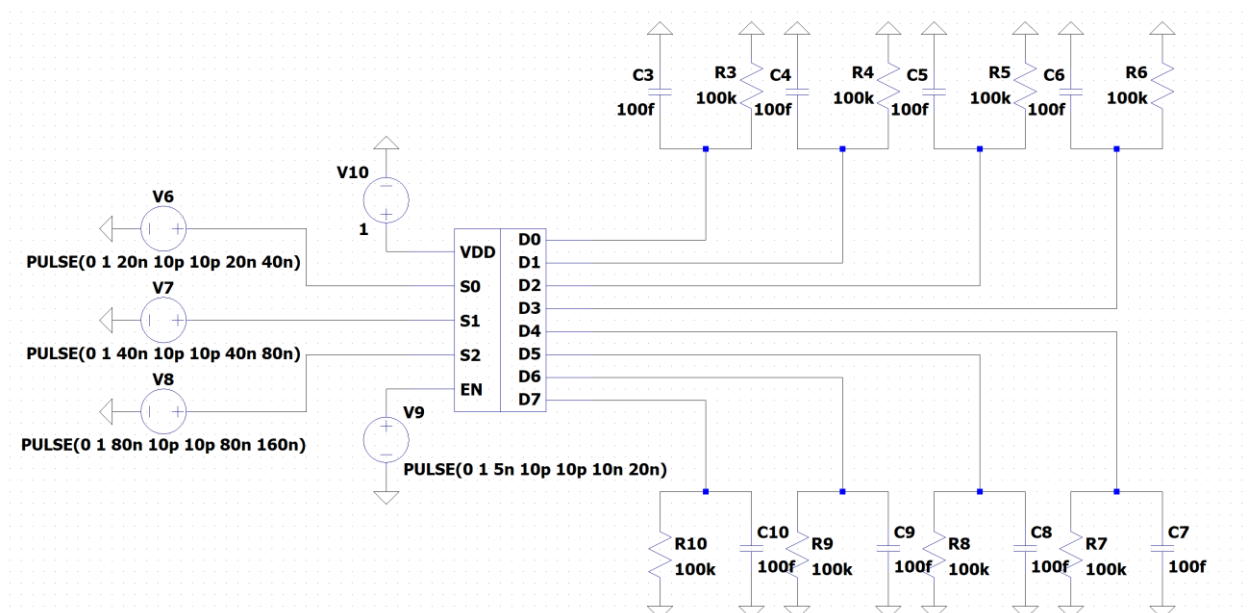


Рисунок 14 – схема тестирования БОЭ

Временная диаграмма процесса тестирования БОЭ:

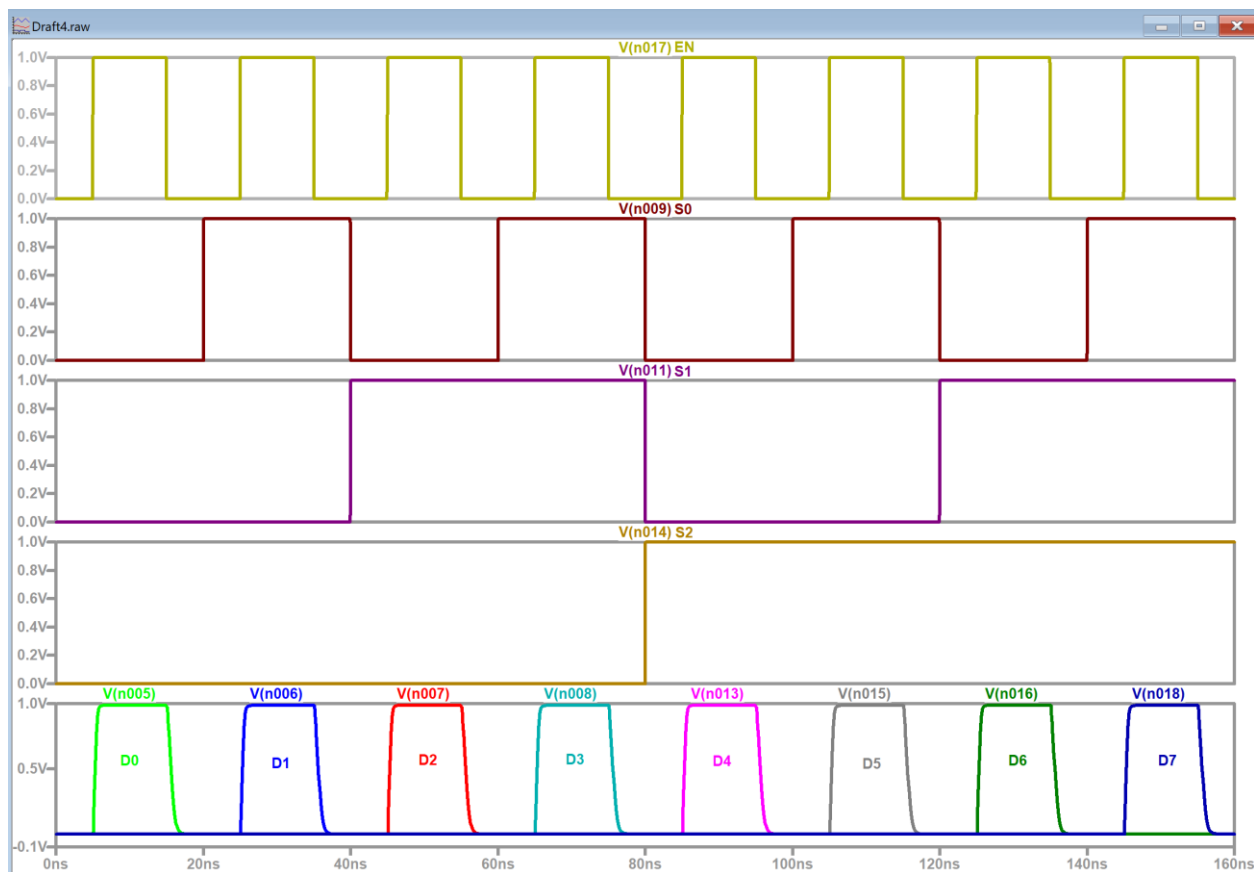


Рисунок 15 – временная диаграмма

Результат измерения задержки распространения сигнала через БОЭ:

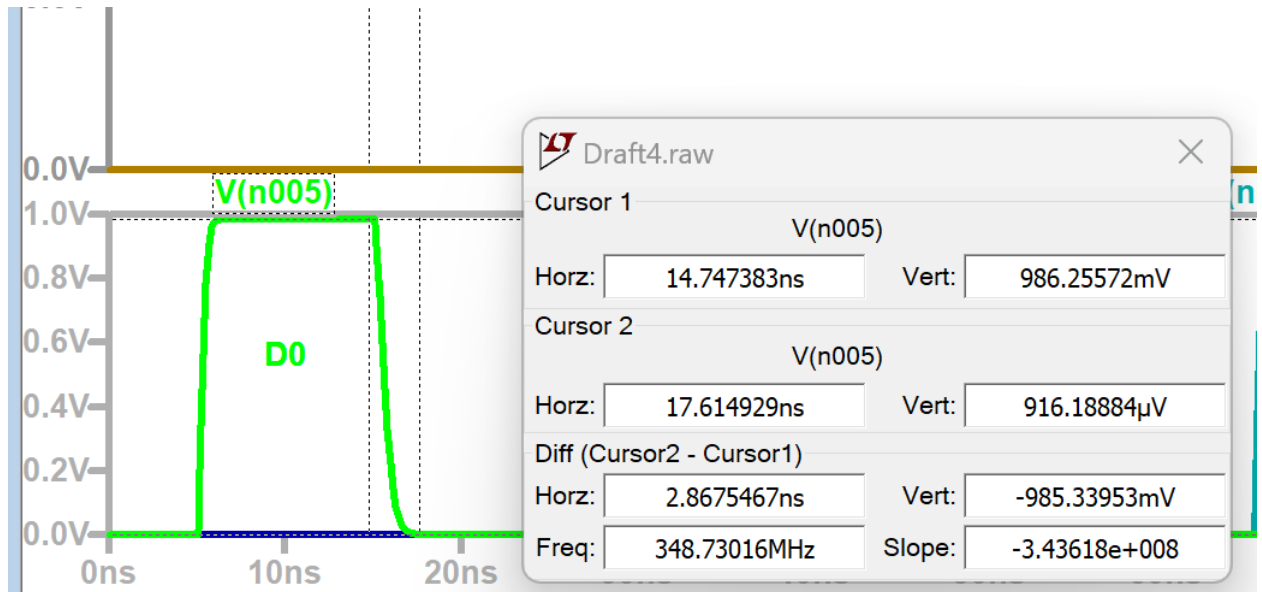


Рисунок 16 - Задержка спада

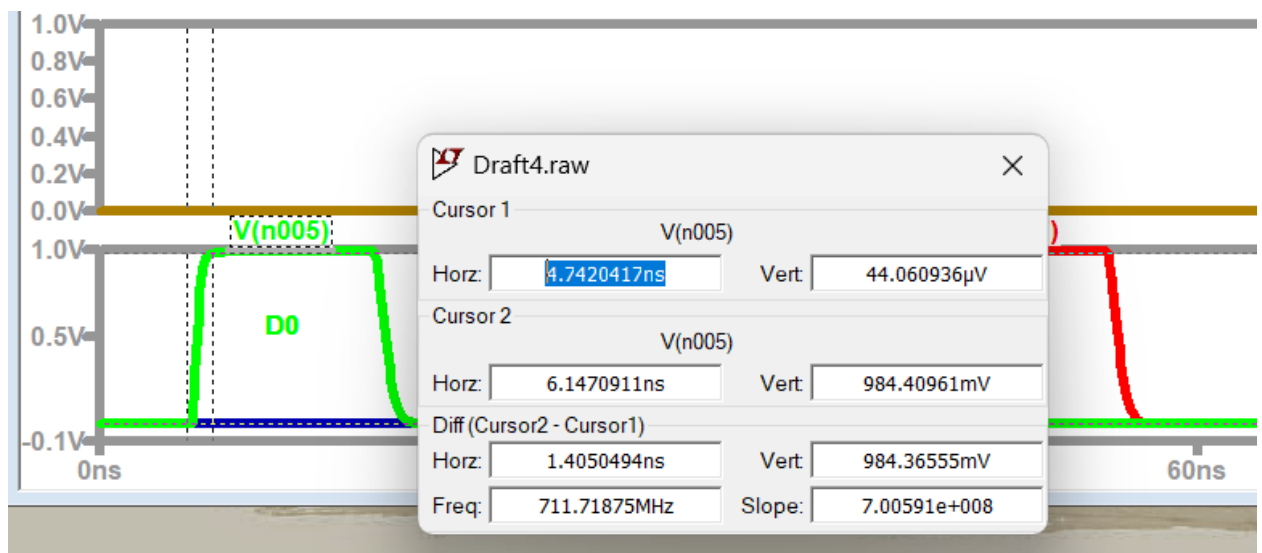


Рисунок 17 - Задержка фронта

Два курсора на верхней (~ 1 В) и нижней (~ 0 В) границах заднего фронта.

Задержка спада равна $T1 \sim 2,87$ нс

Задержка фронта равна $T2 \sim 1,4$ нс

Тогда максимальная частота работы БОЭ равна $f = 1/(T1+T2) = 234$ МГц

Часть № 2. Verilog.

Код разработанного модуля БОЭ:

```
1      | timescale 1ns / 1ps
2
3      module decoder(
4          input [2:0] s,
5          input en,
6          output [7:0] d
7      );
8
9          wire [2:0] not_s;
10         wire [7:0] not_d;
11         wire [7:0] s_2_1;
12         wire [7:0] not_s_2_1;
13         wire [7:0] s_0_en;
14         wire [7:0] not_s_0_en;
15
16         nand(not_s[2], s[2], s[2]);
17         nand(not_s[1], s[1], s[1]);
18         nand(not_s[0], s[0], s[0]);
19
20         nand(not_s_2_1[0], not_s[2], not_s[1]);
21         nand(not_s_0_en[0], not_s[0], en);
22         nand(s_2_1[0], not_s_2_1[0], not_s_2_1[0]);
23         nand(s_0_en[0], not_s_0_en[0], not_s_0_en[0]);
24         nand(not_d[0], s_2_1[0], s_0_en[0]);
```

Рисунок 18 – код БОЭ

```

25
26     nand(not_s_2_1[1], not_s[2], not_s[1]);
27     nand(not_s_0_en[1], s[0], en);
28     nand(s_2_1[1], not_s_2_1[1],not_s_2_1[1]);
29     nand(s_0_en[1], not_s_0_en[1], not_s_0_en[1]);
30     nand(not_d[1], s_2_1[1], s_0_en[1]);
31
32     nand(not_s_2_1[2], not_s[2], s[1]);
33     nand(not_s_0_en[2], not_s[0], en);
34     nand(s_2_1[2], not_s_2_1[2],not_s_2_1[2]);
35     nand(s_0_en[2], not_s_0_en[2], not_s_0_en[2]);
36     nand(not_d[2], s_2_1[2], s_0_en[2]);
37
38     nand(not_s_2_1[3], not_s[2], s[1]);
39     nand(not_s_0_en[3], s[0], en);
40     nand(s_2_1[3], not_s_2_1[3],not_s_2_1[3]);
41     nand(s_0_en[3], not_s_0_en[3], not_s_0_en[3]);
42     nand(not_d[3], s_2_1[3], s_0_en[3]);
43
44     nand(not_s_2_1[4], s[2], not_s[1]);
45     nand(not_s_0_en[4], not_s[0], en);
46     nand(s_2_1[4], not_s_2_1[4],not_s_2_1[4]);
47     nand(s_0_en[4], not_s_0_en[4], not_s_0_en[4]);
48     nand(not_d[4], s_2_1[4], s_0_en[4]);

```

Рисунок 19 - код БОЭ

```

50      nand(not_s_2_1[5], s[2], not_s[1]);
51      nand(not_s_0_en[5], s[0], en);
52      nand(s_2_1[5], not_s_2_1[5],not_s_2_1[5]);
53      nand(s_0_en[5], not_s_0_en[5], not_s_0_en[5]);
54      nand(not_d[5], s_2_1[5], s_0_en[5]);
55
56      nand(not_s_2_1[6], s[2], s[1]);
57      nand(not_s_0_en[6], not_s[0], en);
58      nand(s_2_1[6], not_s_2_1[6],not_s_2_1[6]);
59      nand(s_0_en[6], not_s_0_en[6], not_s_0_en[6]);
60      nand(not_d[6], s_2_1[6], s_0_en[6]);
61
62      nand(not_s_2_1[7], s[2], s[1]);
63      nand(not_s_0_en[7], s[0], en);
64      nand(s_2_1[7], not_s_2_1[7],not_s_2_1[7]);
65      nand(s_0_en[7], not_s_0_en[7], not_s_0_en[7]);
66      nand(not_d[7], s_2_1[7], s_0_en[7]);
67
68      nand(d[0], not_d[0], not_d[0]);
69      nand(d[1], not_d[1], not_d[1]);
70      nand(d[2], not_d[2], not_d[2]);
71      nand(d[3], not_d[3], not_d[3]);
72      nand(d[4], not_d[4], not_d[4]);
73      nand(d[5], not_d[5], not_d[5]);
74      nand(d[6], not d[6], not d[6]);

```

Рисунок 20 - код БОЭ

```

75      nand(d[7], not_d[7], not_d[7]);
76
77      endmodule

```

Рисунок 21 - код БОЭ

Вход: 3-битная шина s с декодируемым числом и сигнал разрешения en.

Выход: 8-битная шина d, где порядок равного 1 бита соответствует декодированному числу.

Используются 6 вспомогательных (5 6-битных и 1 3-битная) шин для передачи результатов операции NAND.

not_s, not_d – инвертированные биты шин s и d.

(not_)s_2_1[n] - (не)инвертированные биты результата (not_)s[2] NAND (not_)s[1] (т.е. выполнение NAND для соответствующих числу n по инверсии старшего и среднего битов).

(not_)s_0_en[n] - (не)инвертированные биты результата (not_)s[0] NAND (not_)en (т.е. выполнение NAND для соответствующего числу n по инверсии младшего бита и сигнала разрешения).

Можно заметить, что в коде 8 раз повторяются разделённые переносом 5 строчек, которые отличаются только номером бита и присутствием/отсутствием приставки «not_» для нужной для этого числа инверсии.

Код разработанного тестового окружения БОЭ:

```
1    `timescale 1ns / 1ps
2
3    module decoder_tb;
4
5        reg [2:0] s;
6        wire[7:0] d;
7        reg en;
8
9        integer i;
10
11        decoder decoder_1(
12            .s(s),
13            .d(d),
14            .en(en)
15        );
16
17        initial begin
18            for(i = 0; i < 8; i = i+1) begin
19                s = i;
20                en = 1;
21
22                #10
23
24                if (d == 2**i) begin
25                    $display("Correct! s=%b, d=%b, en=%b, i=%0d", s, d, en, i);
26                end else begin
```

Рисунок 22 – Код тестового окружения


```

27         $display("Incorrect! s=%b, d=%b, en=%b, i=%0d", s, d, en, i);
28     end
29
30     en = 0;
31
32     #10
33
34     if (d == 0) begin
35         $display("Correct! s=%b, d=%b, en=%b, i=%0d", s, d, en, i);
36     end else begin
37         $display("Incorrect! s=%b, d=%b, en=%b, i=%0d", s, d, en, i);
38     end
39
40 end
41 #10 $stop;
42
43 end
44 endmodule

```

Рисунок 23 - Код тестового окружения

В тестовом окружении задаются переменные входа (s, en) и выхода (d), а также целочисленная переменная i, которая будет изменяться в цикле от 0 до 8; её значение будет присваиваться шине s. Затем проверяется работоспособность дешифратора при включенном и выключенном сигнале разрешения; так как число на входе соответствует порядку равного 1 бита, можно проверять это как равенство результата степени двойки этого числа.

Временная диаграмма процесса тестирования БОЭ:

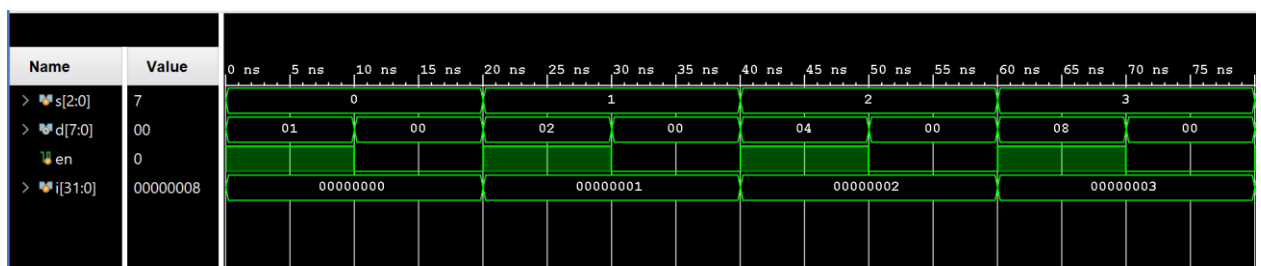


Рисунок 24 - временная диаграмма

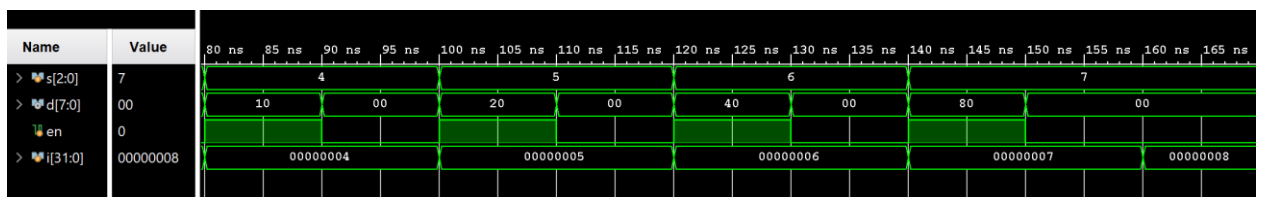


Рисунок 25 - временная диаграмма

Как можно видеть, во время процесса тестирования переменная i изменялась от 0 до 8 каждые 20 нс, переменная s – от 0 до 7 (в последний раз

присваивания не происходило), сигнал en каждые 10 нс инвертировался, вследствие чего переменная d в течение этого промежутка времени менялась от степени двойки числа s к 0 и наоборот.

Вывод в консоль:

Correct! s=000, d=00000001, en=1, i=0

Correct! s=000, d=00000000, en=0, i=0

Correct! s=001, d=00000010, en=1, i=1

Correct! s=001, d=00000000, en=0, i=1

Correct! s=010, d=00000100, en=1, i=2

Correct! s=010, d=00000000, en=0, i=2

Correct! s=011, d=00001000, en=1, i=3

Correct! s=011, d=00000000, en=0, i=3

Correct! s=100, d=00010000, en=1, i=4

Correct! s=100, d=00000000, en=0, i=4

Correct! s=101, d=00100000, en=1, i=5

Correct! s=101, d=00000000, en=0, i=5

Correct! s=110, d=01000000, en=1, i=6

Correct! s=110, d=00000000, en=0, i=6

Correct! s=111, d=10000000, en=1, i=7

Correct! s=111, d=00000000, en=0, i=7

Выводы по работе:

В процессе выполнения данной работы я познакомился со средой Ltspice и языком описания аппаратуры Verilog. В качестве опытного образца я создал собственный вентиль NAND и на его основе создал и протестировал позиционный шифратор «3 в 8».